A) *Listing of the Claims:*

1. (Cancelled)

2. (Currently amended) The method according to claim ~~1~~4, further comprising:

wherein ~~retrieving from the memory comprises retrieving from~~ the memory comprises a read-only memory (ROM).

3. (Currently amended) The method according to claim ~~1~~4 wherein, for serially generating the Gold code sequence, the method further comprising:

~~adding n~~ adding the fixed integer value n and the count value i for each count value i of the plurality of count values.

4. (Currently amended) A method for use in generating one or more data sequences for spread spectrum communications, the method comprising:

serially generating a Gold code sequence by, for each count value i of a plurality of count values:

retrieving from a memory a bit of a pseudorandom noise (PN) sequence corresponding to an (i+n)th position in the PN sequence, where n is a fixed integer value;

retrieving from the memory a bit of the PN sequence corresponding to an (q*i)th position in the PN sequence, where 1 is a fixed integer value;

adding the bid corresponding to the (i+n)th position with the bit corresponding to the (q*i)th position; and

~~A method according to claim 1~~ wherein, for serially generating the Gold code sequence, the method further comprises:

multiplying q and the count value i for each count value i of the plurality of count values.

5. (Currently amended) A method for use in generating one or more data sequences for spread spectrum communications, the method comprising:

serially generating a Gold code sequence by, for each count value i of a plurality of count values:

retrieving from a memory a bit of a pseudorandom noise (PN) sequence corresponding to an (i+n)th position in the PN sequence, where n is a fixed integer value;

retrieving from the memory a bit of the PN sequence corresponding to an (q*i)th position in the PN sequence, where 1 is a fixed integer value;

adding the bid corresponding to the (i+n)th position with the bit corresponding to the (q*i)th position;

~~The method according to claim 1, further comprising:~~

wherein retrieving ~~from memory~~ from the memory a bit of the PN sequence corresponding to the (i+n)th position comprises applying an (i+n) value to the address inputs of the memory; and

wherein retrieving ~~from memory~~ from the memory a bit of the PN sequence corresponding to the (q*i)th position comprises applying an (q*i) value to the address inputs of the memory.

6. (Currently amended) A method for use in generating one or more data sequences for spread spectrum communications, the method comprising:

serially generating a Gold code sequence by, for each count value i of a plurality of count values:

retrieving from a memory a bit of a pseudorandom noise (PN) sequence

corresponding to an (i+n)th position in the PN sequence, where n is a fixed integer value;

retrieving from the memory a bit of the PN sequence corresponding to an (q*i)th position in the PN sequence, where 1 is a fixed integer value;

adding the bid corresponding to the (i+n)th position with the bit corresponding to the (q*i)th position;

~~The method according to claim 1~~, wherein the Gold code sequence is a first Gold code sequence, the method further comprising:

serially generating a second Gold code sequence by, for each count value i of the plurality of count values:

retrieving from memory a bit of the PN sequence corresponding to the (i+n+m)th position in the PN sequence, where "m" is a fixed integer value;

retrieving from memory a bit of the PN sequence corresponding to the (q*i+q*m)th position in the PN sequence; and

adding the bit corresponding to the (i+n+m)th position with the bit corresponding to the (q*i+q*m)th position.

7. (Cancelled)

8. (Cancelled)

9. (Currently amended) In a dual mode Code Division Multiple Access (CDMA), a method for generating an nth Gold code from a pseudorandom noise (PN) sequence stored sequentially ~~in memory~~ in a memory as x(0), x(1), . . . , the method comprising the steps of:

accessing the memory sequentially starting from location ~~"n"~~ n in order to generate the sequence x(i+n), where ~~n"~~ n is a fixed integer value;

accessing the memory non-sequentially starting from a first location (k) and then accessing each qth location in order to generate the sequence x(q*i+k); ~~and~~ where q is a fixed integer value; and

adding on a bit-by-bit basis the resulting two retrieved sequences x(i+n) and x(q*i+k).

10. (Currently amended) A method of generating a complex Gold Code sequence, Z2n(i), applicable to the Universal Mobile Telephone Service (UMTS) standard, where, ~~"x"~~ x is a PN sequence stored sequentially as x(0), x(1), . . . in ~~memory, and x(i) and y(i) are two related sequences each~~ a memory, the PN sequence having a length equal to 2M-1, the method comprising the steps of:

accessing ~~from memory~~ from the memory in sequences of x(i+n+m_, x(q*i+q*m), x(i+n) and x(q*i); and

performing the equation:

$$Z2n(i) = x(i+n) + x(q*i) + j[x(i+n+m) + x(q*i+q*m)]$$

where, "n" and "q" are fixed integer ~~values.~~ values to produce the complex Gold Code sequence, and "M", "n" and "i" are integer values.

11. (Currently amended) A data sequence generator for serially generating one or more data sequences, the data sequence generator comprising:

~~memory~~ a memory;

data stored in said memory;

the data comprising a pseudo-random noise (PN) sequence;

a counting device;

a first adder, including:

a first input coupled to an output of the counting device;

a second input which receives a value n;

a multiplier, including:

a first input coupled to the output of the counting device;

a second input which receives a value q;

a first multiplexer, including:

a first input coupled to an output of the first adder;

a second input coupled to an output of the multiplier; and

an output for coupling to ~~memory address inputs~~ an address input of the memory.

12. (Original) The data sequence generator according to claim 11, further comprising:

the memory comprising a read-only memory (ROM).

13. (Original) The data sequence generator according to claim 11, further comprising:

an output of the memory to provide serially-generated PN sequences responsive to the counting device.

14. (Original) The data sequence generator according to claim 11, further comprising:

a first latch having an input coupled to an output of the memory;

a second latch having an input coupled to the output of the memory;

a second adder, including:

a first input coupled to an output of the first latch;

a second input coupled to an output of the second latch; and

an output to provide a serially-generated Gold code sequence.

15. (Currently amended) The data sequence generator according to claim 11, further comprising:

a second multiplexer, including:

a first input coupled to the output of the first multiplexer;

a second input coupled to the output of the counting device; and

an output coupled to ~~the memory address inputs~~ an address input of the memory.

16. (Currently amended) The data sequence generator according to claim 11, further comprising:

a first latch having an input coupled to an output of the memory;

a second latch having an input coupled to the output of the memory;

a second adder, including:

a first input coupled to an output of the first latch;

a second input coupled to an output of the second latch;

an output to provide a serially-generated Gold code sequence;

a second multiplexer, including:

a first input coupled to the output of the second adder;

a second input coupled to the output of the memory; and

an output to provide, in a time-multiplexed ~~fashion, a~~ fashion the serially-generated PN sequence and the serially-generated Gold code sequence.

17. (Currently amended) The data sequence generator according to claim 11, further comprising:

a second multiplexer, including:

a first input coupled to the output of the first multiplexer;

a second input coupled to the output of the counting device;

an output coupled to ~~the memory address inputs~~ an address input of the memory;

a first latch having an input coupled to an output of the memory;

a second latch having an input coupled to the output of the memory;

a second adder, including:

a first input coupled to an output of the first latch;

a second input coupled to an output of the second latch;

an output to provide a serially-generated Gold code sequence;

a third multiplexer, including:

a first input coupled to the output of the second adder;

a second input coupled to the output of the memory; and

an output to provide, in a time-multiplexed fashion, a serially-generated PN sequence and a serially-generated Gold code sequence.

18. (Currently amended) A data sequence generator, comprising:

a read-only memory (ROM) storing a pseudo-random noise (PN) sequence;

a counter;

a first adder, including:

a first input coupled to the output of the counter;

a second input which receives a value n;

a multiplier, including:

a first input coupled to the output of the counter;

a second input which receives a value q;

a first multiplexer, including:

a first input coupled to an output of the first adder;

a second input coupled to an output of the multiplier;

a second multiplexer, including:

a first input coupled to an output of the first multiplexer;

a second input coupled to the output of the counter; and

an output coupled to ~~memory address inputs~~ an address input of the ROM.

19. (Original) The data sequence generator according to claim 18, further comprising:

a first latch coupled to an output of the ROM;

a second latch coupled to the output of the ROM;

a second adder, including:

a first input coupled to an output of the first latch;

a second input coupled to an output of the second latch; and

an output to provide a Gold Code sequence.

20. (Original) The data sequence generator according to claim 18, further comprising:

a first latch coupled to an output of the ROM;

a second latch coupled to the output of the ROM;

a second adder, including:

a first input coupled to an output of the first latch;

a second input coupled to an output of the second latch;

a third multiplexer, including:

a first input coupled to the output of the ROM;

a second input coupled to an output of the second adder; and

an output to selectively provide the PN sequence and a Gold Code sequence.

21. (Currently amended) The data sequence generator according to claim 20, wherein the ROM comprises a first ~~read-only memory (ROM)~~ ROM and a second ROM and the output of the second multiplexer is coupled to memory address inputs of both the first and the second ROM;

a first PN sequence is stored in the first ROM and a second PN sequence is stored in the second ROM;

the first and second latches are coupled to the output of the first ROM; and

the first input of the third multiplexer is coupled to the output of the second ROM.

22. (Currently amended) A data sequence generator for use in direct sequence spread spectrum (DSSS) communications, comprising:

memory;

a pseudo-random noise (PN) sequence stored in the memory;

a counter for use in generating each count value i of a plurality of count values;

an output of the memory to provide, for each count value I received at

memory address inputs, a bit of the PN sequence corresponding to the (i)th position in the PN sequence;

an output of the memory to provide, for each (i+n) value received at the memory address inputs, a bit of the PN sequence corresponding to the (i+n)th position in the PN ~~sequence;~~ sequence, where n is a fixed integer value;

an output of the memory to provide, for each (q*i) value received at the memory address inputs, a bit of the PN sequence corresponding to the (q*i)th position in the PN ~~sequence;~~ sequence, where q is a fixed integer value; and

an adder to provide a sum of the bit corresponding to the (i+n)th position and the bit corresponding to the (q*i)th position, to thereby provide a Gold code sequence.